DOCKET: TUC9-2000-0023-US1

While the preferred embodiments of the present invention
have been illustrated in detail, it should be apparent that
modifications and adaptations to those embodiments may occur to
one skilled in the art without departing from the scope of the
5 present invention as set forth in the following claims.

I claim:

DOCKET: TUC9-2000-0023-US1

1.   A method for generating a redundant, updatable, self-booting

program in a sectored, non-volatile memory, said memory having

separately erasable sectors, comprising the steps of:

writing said memory with a boot program in at least one of

5 said sectors;

writing said memory with at least two copies of operational

code, each said operational code copy written in at least one of

said sectors separate from said boot sector and separate from

other of said copy sectors, each said operational code copy

10 comprising information indicating the relative update level of

said operational code copy, whereby said copies may be separately

updated and have different update levels;

said boot program reading said update level information to

determine the most recent said update level of said operational

15 code copies; and

said boot program executing said operational code copy

having said most recent update level.

2.   The method of Claim 1, additionally comprising the steps of:

prior to said update level information reading step, said

20 boot program testing said operational code copies to determine

whether any of said operational code copies is corrupted; and

18

DOCKET: TUC9-2000-0023-US1

conducting said update level information reading step of
only those of said copies determined to be uncorrupted in said
boot program testing step.

3.   The method of Claim 2, wherein said testing step comprises
5 conducting a checksum test of each said operational code copy and
testing the checksum remainder for errors.

4.   The method of Claim 1, additionally comprising the steps of:
an update routine reading said update level information;
said update routine determining from said read update level
10 information the least recent said update level of said
operational code copies; and
said update routine updating said operational code copy
having said least recent update level.

5.   The method of Claim 4, wherein said updating routine
15 updating step comprises erasing and updating only said sectors
having said programmed operational code copy.

DOCKET: TUC9-2000-0023-US1

6.   The method of Claim 4, additionally comprising the steps of:

    said update routine reading said update level information to determine whether any of said copies has an update level less recent than said copy of said updating step; and

5   if any of said copies has said less recent update level, said update routine repeating said least recent determining and operational code updating steps, until all of said copies are at said update level of said copy of said updating step.

7.   The method of Claim 1, additionally comprising the steps of:

10   an update routine testing said operational code copies to determine whether any of said operational code copies is corrupted; and

    said update routine erasing said corrupted operational code copy and overwriting said copy with an updated operational code

15 copy.

8.   The method of Claim 7, wherein said updating routine erasing and overwriting step comprises erasing and overwriting only said sectors having said corrupted operational code copy.

20

DOCKET: TUC9-2000-0023-US1

9.   The method of Claim **7**, wherein said testing step comprises conducting a checksum test of each said operational code copy and testing the checksum remainder for errors.

10.   The method of Claim **7**, additionally comprising the steps of:

5   subsequent to said update routine testing step and any said update routine erasing and overwriting steps, said update routine reading said update level information;

said update routine determining from said read update level information the least recent said update level of said

10 operational code copies; and

said update routine updating said operational code copy having said least recent update level.

11.   The method of Claim **1**, additionally comprising the steps of:

an update routine contained in an input update code image

15 updating said operational code copies having a less recent update level than said input update code image.

21

DOCKET: TUC9-2000-0023-US1

12. A system for providing a redundant, updatable, self-booting
program, comprising:

a sectored, non-volatile memory, having separately erasable
sectors, said memory storing at least two copies of operational
5 code, each said operational code copy stored in at least one of
said sectors separate from other of said copy sectors, each said
operational code copy having information indicating the relative
update level of said operational code copy, whereby said copies
may be separately updated and have different update levels;

10   a boot processor, employing a boot program stored in at
least one of said sectors of said non-volatile memory, separate
from said operational code copy sectors, said boot processor
comprising a microprocessor operated to read said update level
information to determine the most recent said update level of
15 said operational code copies, and said boot processor operating
said microprocessor to execute said operational code copy having
said most recent update level.

22

DOCKET: TUC9-2000-0023-US1

13. The system of Claim **12**, wherein said boot processor,

additionally:

prior to reading said update level information, testing said

operational code copies to determine whether any of said

5 operational code copies is corrupted; and

reading said update level information of only those of said

copies determined to be uncorrupted in said testing.

14. The system of Claim **13**, wherein said boot processor conducts

said testing, conducting a checksum test of each said operational

10 code copy and testing the checksum remainder for errors.

15. The system of Claim **12**, additionally comprising an update

processor comprising code operating a microprocessor to update

said operational code copies: reading said update level

information; determining from said read update level information

15 the least recent said update level of said operational code

copies; and updating said operational code copy having said least

recent update level with an update code image.

16. The system of Claim **15**, wherein said update processor

updates only said sectors of said non-volatile memory having said

20 least recent update level operational code copy.

23

DOCKET: TUC9-2000-0023-US1

17. The system of Claim **15,** wherein said update processor additionally reads said update level information to determine whether any of said copies has an update level less recent than said update code image, and, if any of said copies has said less

5 recent update level, updating said operational copies having said less recent update level, until all of said copies are at said update level of said update copy.

18. The system of Claim **15,** wherein said update processor additionally tests said operational code copies to determine

10 whether any of said operational code copies is corrupted, requesting the erasure of any said corrupted operational code copy and overwriting said copy with an updated operational code image.

19. The system of Claim **18,** wherein said update processor

15 conducts said testing, conducting a checksum test of each said operational code copy and testing the checksum remainder for errors.

24

DOCKET: TUC9-2000-0023-US1

20. The system of Claim **18**, wherein said boot processor, subsequent to said testing and any updating, erasing and overwriting: reads said update level information, determining from said read update level information the most recent said

5 update level of said operational code copies, and operates said microprocessor to execute said operational code copy having said most recent update level.

21. The system of Claim **12**, additionally comprising an update processor comprising update routine operating a microprocessor to

10 update said operational code copies, and wherein said update processor routine is incorporated in an input update code image for updating said operational code copies.

DOCKET: TUC9-2000-0023-US1

22. A redundant, updatable, self-booting computer program
product usable with a programmable computer having computer
readable program code embodied therein, for use with a sectored,
non-volatile memory, said memory having separately erasable
5 sectors, said memory having at least two copies of operational
code, each said operational code copy in at least one of said
sectors separate from other of said copy sectors, each said
operational code copy having information indicating the relative
update level of said operational code copy, whereby said copies
10 may be separately updated and have different update levels,
comprising;

computer readable program code which causes a computer
processor to read said update level information to determine the
most recent said update level of said operational code copies;
15 and

computer readable program code which causes a computer
processor to execute said operational code copy having said most
recent update level.

26

DOCKET: TUC9-2000-0023-US1

23. The computer program product of Claim **22**, additionally comprising computer readable program code which causes a computer processor to, prior to reading said update level information, test said operational code copies to determine whether any of

5 said operational code copies is corrupted; and to read said header of only those of said copies determined to be uncorrupted in said test.

24. The computer program product of Claim **23**, wherein said computer readable program code which causes a computer processor

10 to test said operational code copies, causes said computer processor to conduct a checksum test of each said operational code copy and test the checksum remainder for errors.

25. The computer program product of Claim **22**, additionally comprising computer readable program code which causes a computer

15 processor to read said update level information; to determine from said read update level information the least recent said update level of said operational code copies; and to update said operational code copy having said least recent update level.

27

DOCKET: TUC9-2000-0023-US1

26. The computer program product of Claim **25,** wherein said computer readable program code which causes a computer processor to update said operational code copy having said least recent update level causes said computer processor to erase and update

5 only said sectors having said least recent update level operational code copy.

27. The computer program product of Claim **25,** additionally comprising computer readable program code which causes a computer processor to read said update level information to determine

10 whether any of said copies has an update level less recent than an update code image, and if any of said copies has said less recent update level, to determine from said read update level information the least recent said update level of said less recent operational code copies, and update said operational code

15 copy having said least recent update level, until all of said copies is at said update level of said update code image.

DOCKET: TUC9-2000-0023-US1

28. The computer program product of Claim **22**, additionally comprising computer readable program code which causes a computer processor to test said operational code copies to determine whether any of said operational code copies is corrupted, to

5 erase said corrupted operational code copy, and to overwrite said copy with an updated operational code image.

29. The computer program product of Claim **28**, wherein said computer readable program code which causes a computer processor to erase and overwrite said corrupted operational code copy

10 having said least recent update level causes said computer processor to erase and overwrite only said sectors having said corrupted operational code copy.

30. The computer program product of Claim **28**, wherein said computer readable program code which causes a computer processor

15 to test said operational code copies, causes said computer processor to conduct a checksum test of each said operational code copy and test the checksum remainder for errors.

DOCKET: TUC9-2000-0023-US1

31. The computer program product of Claim **28**, additionally

comprising computer readable program code which causes a computer

processor, subsequent to said testing and any said erasing and

overwriting, to read said update level information, to determine

5 from said read update level information the least recent said

update level of said operational code copies, and to update said

operational code copy having said least recent update level.

32. The computer program product of Claim **22**, additionally

comprising update computer readable program code which causes a

10 computer processor to respond to an update code image and update

any said operational code copy having a less recent update level

than said update code image, said update computer readable

program code contained in said update code image.

30